# CWE Detail – CWE-1426

## Description

The product invokes a generative AI/ML
 component whose behaviors and outputs cannot be directly
 controlled, but the product does not validate or
 insufficiently validates the outputs to ensure that they
 align with the intended security, content, or privacy
 policy.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2024-3402: chain: GUI for ChatGPT API performs
 input validation but does not properly "sanitize"
 or validate model output data (CWE-1426), leading
 to XSS (CWE-79).

## Modes of Introduction

**•** Architecture and Design: Developers may rely heavily on protection mechanisms such as
input filtering and model alignment, assuming they are more effective
than they actually are.

**•** Implementation: Developers may rely heavily on protection mechanisms such as
input filtering and model alignment, assuming they are more effective
than they actually are.

## Common Consequences

**•** Impact: Execute Unauthorized Code or Commands, Varies by Context — Notes:

## Potential Mitigations

**•** Architecture and Design: Since the output from a generative AI component (such as an LLM) cannot be trusted, ensure that it operates in an untrusted or non-privileged space. (Effectiveness: N/A)

**•** Operation: Use "semantic comparators," which are mechanisms that
 provide semantic comparison to identify objects that might appear
 different but are semantically similar. (Effectiveness: N/A)

**•** Operation: Use components that operate
 externally to the system to monitor the output and
 act as a moderator. These components are called
 different terms, such as supervisors or
 guardrails. (Effectiveness: N/A)

**•** Build and Compilation: During model training, use an appropriate variety of good
 and bad examples to guide preferred outputs. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Notes

**•** Research Gap: This entry is related to AI/ML, which is not well
 understood from a weakness perspective. Typically, for
 new/emerging technologies including AI/ML, early
 vulnerability discovery and research does not focus on
 root cause analysis (i.e., weakness identification). For
 AI/ML, the recent focus has been on attacks and
 exploitation methods, technical impacts, and mitigations.
 As a result, closer research or focused efforts by SMEs
 is necessary to understand the underlying weaknesses.
 Diverse and dynamic terminology and rapidly-evolving
 technology further complicate understanding. Finally,
 there might not be enough real-world examples with
 sufficient details from which weakness patterns may be
 discovered. For example, many real-world vulnerabilities
 related to "prompt injection" appear to be related to
 typical injection-style attacks in which the only
 difference is that the "input" to the vulnerable
 component comes from model output instead of direct
 adversary input, similar to "second-order SQL injection"
 attacks.

**•** Maintenance: This entry was created by members
 of the CWE AI Working Group during June and July 2024. The
 CWE Project Lead, CWE Technical Lead, AI WG co-chairs, and
 many WG members decided that for purposes of timeliness, it
 would be more helpful to the CWE community to publish the
 new entry in CWE 4.15 quickly and add to it in subsequent
 versions.